Registration of Point Sets with Rigid Transformations

Thomas Funkhouser
Princeton University
CS526, Fall 2010

Point Set Matching

Goal is to compute a similarity measure for a pair of attributed point sets

Distance(A, B) = ?

Point Set Matching

Challenge is to find corresponding points
- "Subset of points in A" may match "subset of points in B"

Point Set Matching

Calculating a superposition and distance measure is easy if correspondences are known (proposed)

Least-squares optimal superposition of corresponding points
Point Set Matching

Calculating a superposition and distance measure is easy if correspondences are known (proposed)

\[ \text{RMSD}(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \]

Distance(A,B) = RMSD(A,B) + OtherTerms …

Outline

Introduction
Point set matching
  - Brute force search
    - RANSAC
    - Geometric hashing
    - Association graphs
    - Iterative closest point
Discussion

Brute Force Search

Simple method:
  - Try all possible sets of point correspondences
  - Score the alignment for each one

Problem:
  - \( O(n^{m^2}) \) possible sets of \( m \) correspondences among \( n \) points

Brute Force Search

Simple method:
  - Try all possible sets of point correspondences
  - Score the alignment for each one

Problem:
  - \( O(n^{m^2}) \) possible sets of \( m \) correspondences among \( n \) points

Brute Force Search

Simple method:
  - Try all possible sets of point correspondences
  - Score the alignment for each one (e.g., RMSD)

Problem:
  - \( O(n^{m^2}) \) possible sets of \( m \) correspondences among \( n \) points

Outline

Introduction
Point set matching
  - Brute force search
    - RANSAC
    - Geometric hashing
    - Association graphs
    - Iterative closest point
Discussion
**RANSAC**

Randomly sample set of possible correspondences
- Randomly generate a small set of point correspondences
- Compute the aligning transformation for correspondences
- Score how well other points align after that transformation

**Outline**

Introduction
- Point set matching
  - Brute force search
  - RANSAC
    - Geometric hashing
    - Association graphs
  - Iterative closest point
- Discussion

**Geometric Hashing**

Discretize transformations and scoring

Point Set A

Point Set B
### Geometric Hashing

#### Discretize transformations and scoring

Point Set A

Point Set B

Score correspondences

#### Query processing

Point Set Query

#### Preprocessing complexity

- \(O(n^4)\) for \(n\) points per binding site
- \(\leq O(n^2)\) possible triples * \(O(n)\) transformations per triple

#### Query complexity

- \(O(m) * \text{binsize for } m \text{ points in query binding site}\)
- \(\leq 1 \text{ triple } * O(m)\) transformations per triple
- binsize hash processing per transformation

### Outline

#### Introduction

Point set matching
- Brute force search
- RANSAC
- Geometric hashing
  - Association graphs
- Iterative closest point

Discussion
Association Graphs

Create vertices in the association graph for all compatible pairs of vertices in the original graphs. This can lead to a large number of vertices.

Create edges between (uv) and (wx) if the edges between (u) and (w) as well as between (v) and (x) match.

[Schmitt, Brown]
Association Graphs

Create edges between (uv) and (wx) if the edges between (u) and (w) as well as between (v) and (x) match.

For this example, edge length is the only consideration.

Finding correspondences: The largest set of corresponding nodes in the same configuration is the maximal clique in the association graph.

Computational complexity:
- $O(2^n)$ for n points
- NP-complete
- Branch and bound algorithms

Outline

Introduction
Point set matching
- Brute force
- RANSAC
- Geometric hashing
- Association graphs
- Iterative closest points

Discussion

Iterative Closest Points

Given two point sets
Iterative Closest Points

Given two point sets

A

B

[Besl92]

Iterative Closest Points

Given two point sets and an initial guess for the transformation that aligns them

[Besl92]

Iterative Closest Points

Assume closest points correspond

[Besl92]

Iterative Closest Points

Assume closest points correspond: A→B

[Besl92]

Iterative Closest Points

Assume closest points correspond: A→B and B→A

[Besl92]

Iterative Closest Points

Rejecting outliers

[Besl92]
**Iterative Closest Points**

Find the transformation that optimally aligns proposed correspondences (superposition)

\[ d(A, B) = \sum_{a \in A} ||a - B||^2 + \sum_{b \in B} ||a - B||^2 \]

**Iterative Closest Points**

Iterate until convergence

1. Select source points (from one or both molecules)
2. Match to points in the other molecule
3. Weight the correspondences
4. Reject outlier point pairs
5. Compute an error metric for the current transform
6. Minimize the error metric w.r.t. transformation

**Computational complexity**
- \( O(k \times \log n) \) for \( n \) points per binding site and \( k \) iterations
- \( 5 \) \( k \) iterations * \( O(n) \) points * \( O(\log n) \) to find closest point

**Summary**

- Brute force
  - Accurate, slow
- RANSAC
  - Approximate
- Geometric hashing
  - Fast query, after slow preprocessing
  - Distance threshold implicit in hash bucket sizes
- Association graphs
  - Expensive for large point sets
  - Distance threshold for “associations”
- Iterative closest points
  - Fast, in practice
  - Requires good initial guess

Demo courtesy of Szymon Rusinkiewicz